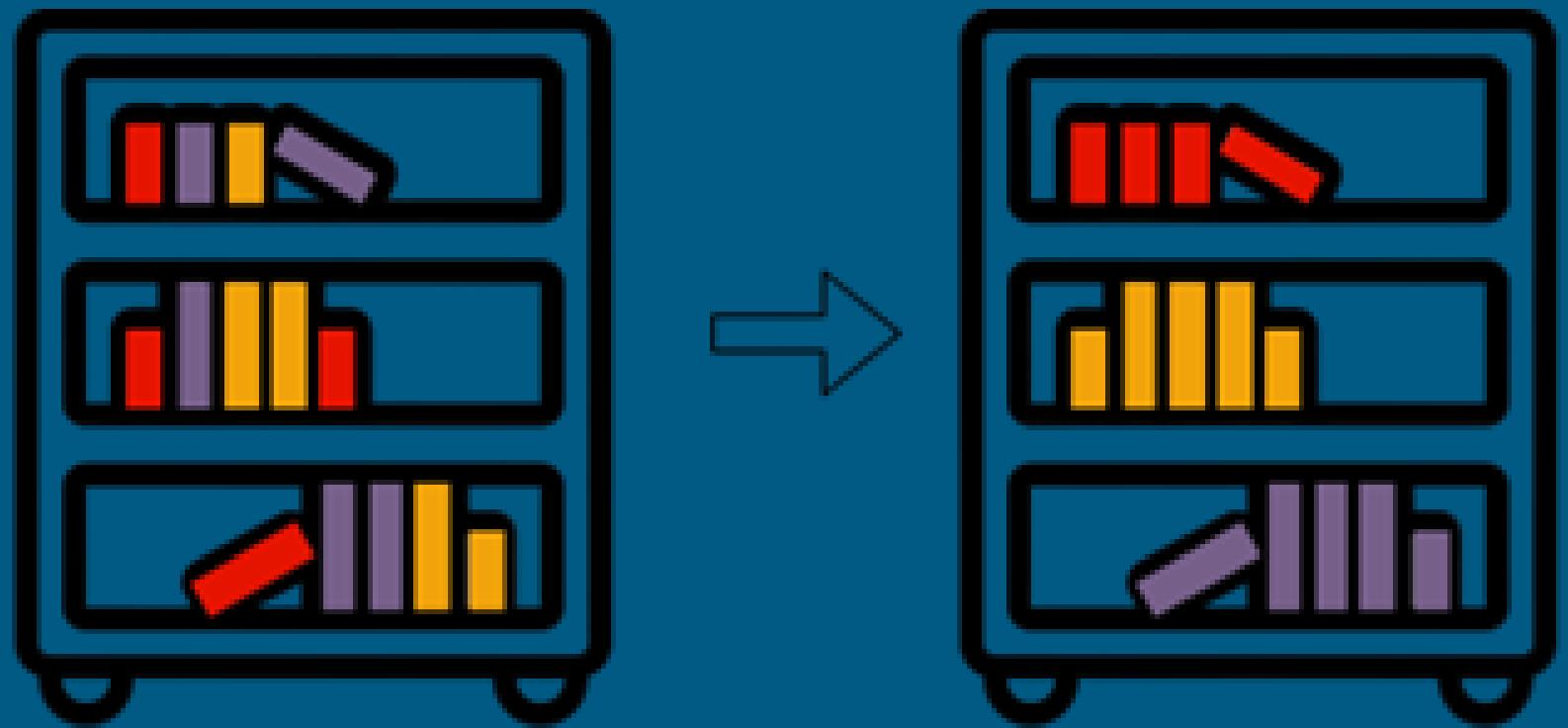


---

**Curso Superior de Tecnologia em Análise e  
Desenvolvimento de Sistemas  
Prof. Edécio Fernando Iepsen**

# **Algoritmos e Estruturas de Dados I**

**AGRUPAMENTOS**



# Agrupamentos: Motivação

Por que agrupar dados:

- Grandes volumes de dados dificultam a análise direta
- Agrupar permite identificar padrões, contar ocorrências e obter resumos estatísticos
- Exemplo do mundo real:
  - Vendas por categoria
  - Alunos por curso
  - Chamados de suporte por status

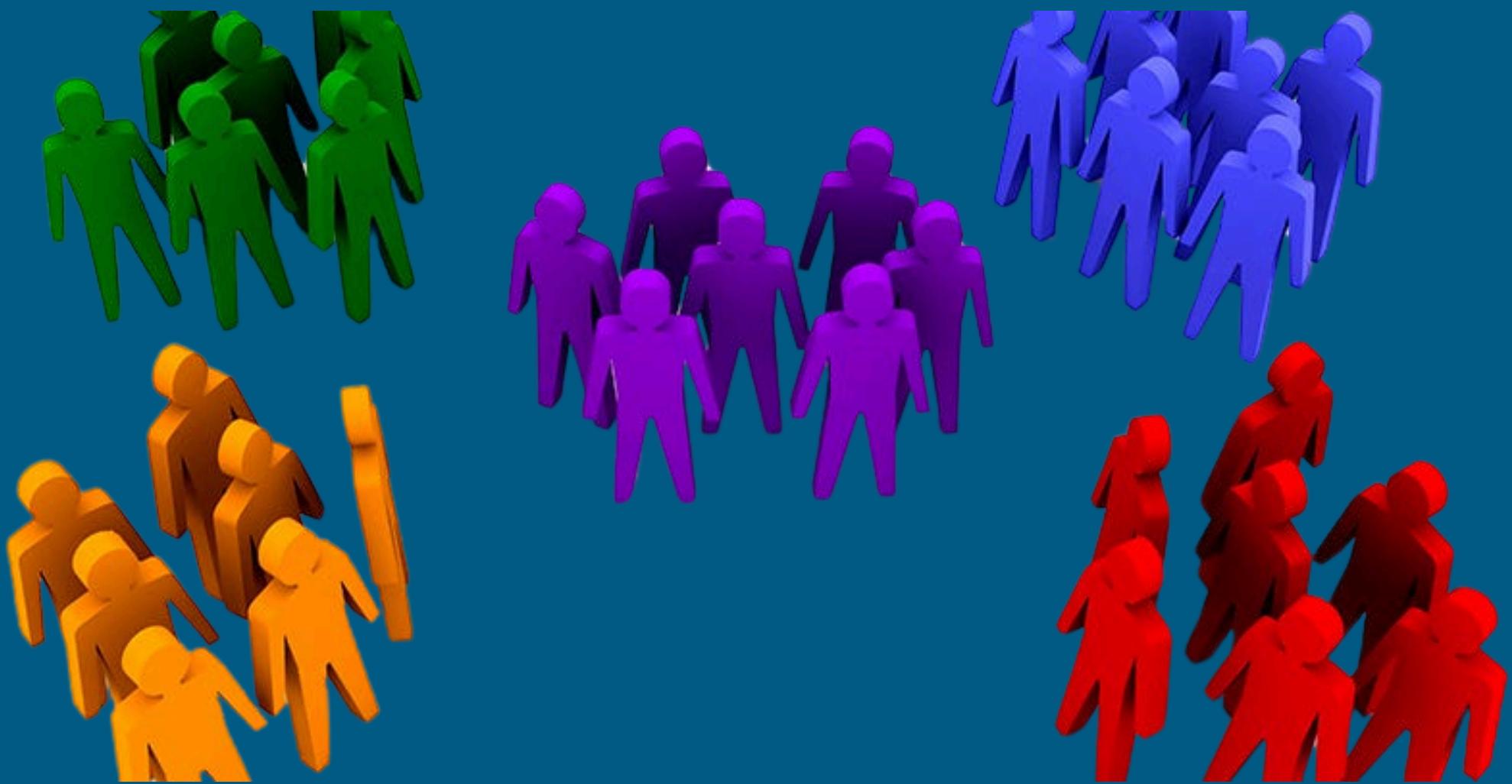
# Exemplo (somar valores):

```
vendas = [  
    {"produto": "Camiseta", "categoria": "Roupas", "valor": 50},  
    {"produto": "Calça", "categoria": "Roupas", "valor": 120},  
    {"produto": "Notebook", "categoria": "Eletrônicos", "valor": 2500},  
    {"produto": "Mouse", "categoria": "Eletrônicos", "valor": 100},  
]  
  
grupos = {}  
  
for venda in vendas:  
    cat = venda["categoria"]  
    valor = venda["valor"]  
    # grupos.get() obtém o valor da chave pesquisada ou 0 se não existir  
    grupos[cat] = grupos.get(cat, 0) + valor  
  
print(grupos)  
# Saída: {'Roupas': 170, 'Eletrônicos': 2600}
```

# Exemplo (contar nº de ocorrências):

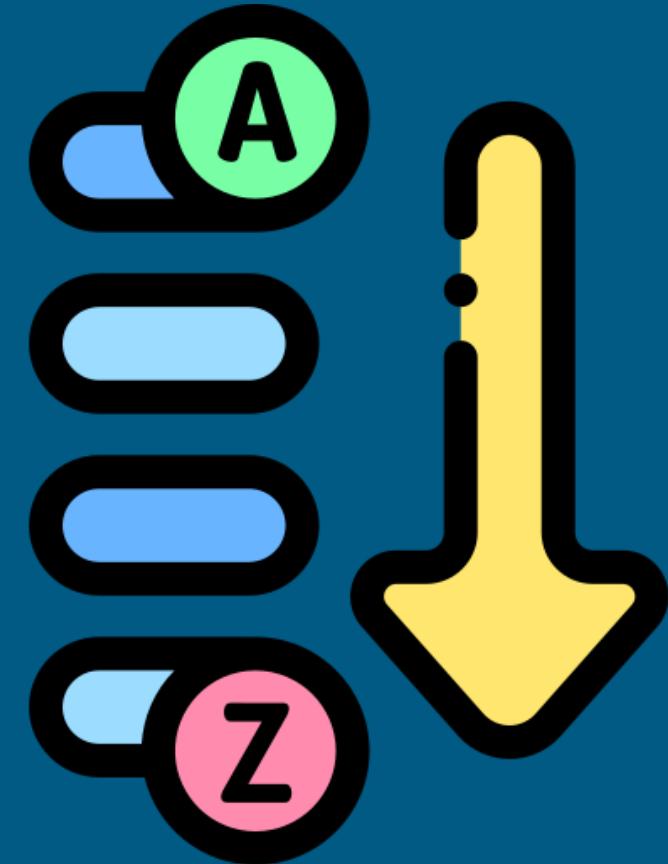
```
vendas = [  
    {"produto": "Camiseta", "categoria": "Roupas", "valor": 50},  
    {"produto": "Calça", "categoria": "Roupas", "valor": 120},  
    {"produto": "Notebook", "categoria": "Eletrônicos", "valor": 2500},  
    {"produto": "Mouse", "categoria": "Eletrônicos", "valor": 100},  
]  
  
grupos = {}  
  
for venda in vendas:  
    cat = venda["categoria"]  
    grupos[cat] = grupos.get(cat, 0) + 1  
  
print(grupos)  
# Saída: {'Roupas': 2, 'Eletrônicos': 2}
```

# Benefícios do Agrupamento



- ✓ Redução de complexidade
- ✓ Facilidade na visualização e análise
- ✓ Apoia decisões baseadas em dados
- ✓ Preparação para visualizações gráficas (pandas, matplotlib etc.)

# Ordenação:



## ✓ Melhor visualização

Resultados agrupados ficam mais fáceis de ler e interpretar



## Preparação para exibição

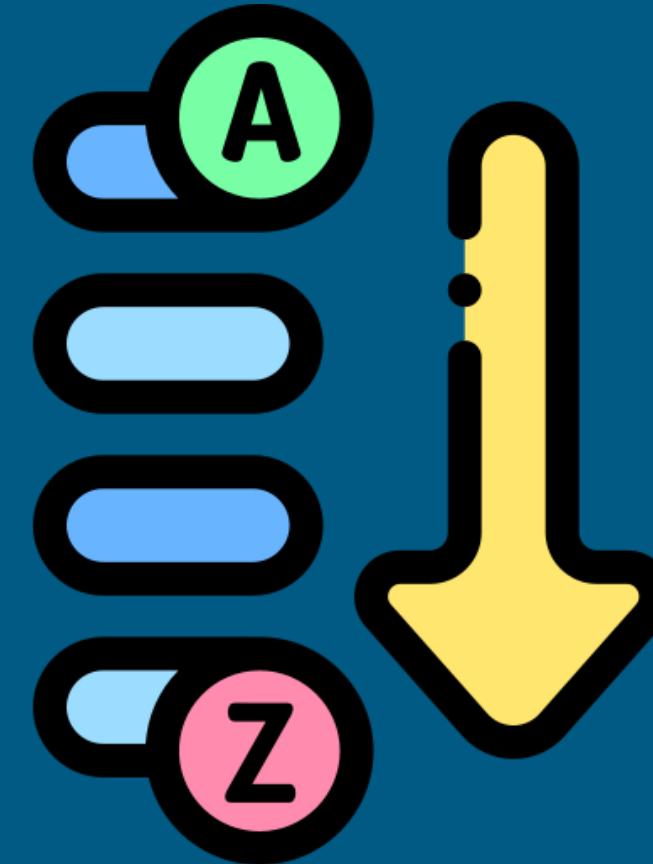
Tabelas e gráficos ficam mais organizados quando os dados seguem uma ordem lógica



## Facilita análises

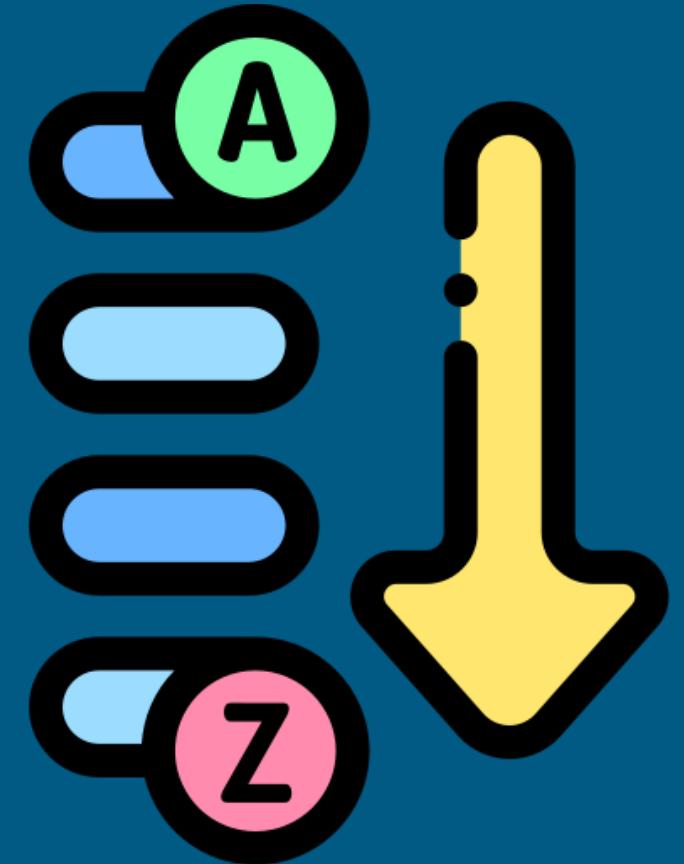
Ao ordenar por valor, é possível identificar rapidamente os grupos de maior impacto

# Ordenação:



```
# Ordenar por categoria  
grupos = {}  
  
for venda in vendas:  
    cat = venda["categoria"]  
    valor = venda["valor"]  
    grupos[cat] = grupos.get(cat, 0) + valor  
  
# Ordenar por chave  
grupos2 = dict(sorted(grupos.items()))  
  
print(grupos2)  
# Saída: {'Eletrônicos': 2600, 'Roupas': 170}
```

# Ordenação:



```
# Ordenar por valor (decrescente)
grupos = {}

for venda in vendas:
    cat = venda["categoria"]
    valor = venda["valor"]
    grupos[cat] = grupos.get(cat, 0) + valor

# Ordenar por valor
grupos3 = dict(sorted(grupos.items(),
                      key=lambda grupo: grupo[1], reverse=True))

print(grupos3)
# Saída: {'Eletrônicos': 2600, 'Roupas': 170}
```

# Conclusões:

- 📌 **Agrupar dados = sintetizar conhecimento**
- 🧠 Permite extrair informações valiosas de grandes listas
- 👨‍🏫 Essencial para quem trabalha com análise, relatórios e dashboards

# Exercícios:

Dado o arquivo (obtido da plataforma Kaggle) com a lista dos pilotos vencedores de corridas da Fórmula 1, obtenha:

- A lista dos 10 pilotos com maior número de vitórias na história da F1 (em ordem decrescente de vitórias)
- A lista das Equipes com 10 ou + vitórias na F1 (em ordem crescente de vitórias)
- A lista das 10 provas com vitórias com maior tempo da história da F1 (em ordem decrescente de tempo)
  - Tempo, Piloto, Equipe, GP e Data

